Statistical physics of learning vector quantization

Witoelar, Aree Widya

Material based on: A. Witoelar, M. Biehl, A. Ghosh and B. Hammer – *"Learning Dynamics of Neural Gas and Vector Quantization,"* Neurocomputing 71, pp. 1210–1219, 2008.

Chapter 3

Learning dynamics and robustness of vector quantization and neural gas

Abstract

Various alternatives have been developed to improve the Winner-Takes-All (WTA) mechanism in vector quantization, including the Neural Gas (NG). However, the behavior of these algorithms including their learning dynamics, robustness with respect to initialization, asymptotic results, etc. has only partially been studied in a rigorous mathematical analysis. The theory of on-line learning allows for an exact mathematical description of the training dynamics in model situations. We analyse both WTA and NG schemes using a system of three competing prototypes trained from a mixture of Gaussian clusters and demonstrate that the Neural Gas can improve convergence speed and achieves robustness to initial conditions. However, depending on the structure of the data, the Neural Gas does not always obtain the best asymptotic quantization error.

3.1 Introduction

Vector quantization (VQ) as developed by Kohonen in (Kohonen 1997) is an important unsupervised learning algorithm, widely used in different areas such as data mining, medical analysis, image compression and speech or handwriting recognition. An up-to-date repository of applications can be found in (Neural Networks Research Centre, Helsinki 2002). The main objective of VQ is to represent the data points by a small number of prototypes or codebook vectors, measured by a distortion or quantization error. This can directly be used for compression, clustering, data mining or (with post-labeling of the prototypes) classification (Gersho and Gray 1991, Jain et al. 1999).

The basic "winner-takes-all" (WTA) or batch algorithms such as the popular k-means clustering (Bottou and Bengio 1995) directly optimize the quantization error underlying vector quantization. However, these methods can easily be subject to confinement in local minima of the quantization error and produce suboptimal results. Consequently, the initialization of prototypes undesirably plays a critical

role in the success of training. A variety of alternatives to overcome this problem has been proposed, some of which are heuristically motivated while others are based on the minimization of a cost function related to the quantization error: the self-organizing map (SOM) (Kohonen 1997), fuzzy-k-means (Bezdek 1981), stochastic optimization (Buhmann 1998), to name just a few. These algorithms have in common that each pattern influences more than one prototype at a time through a "winner-takes-most" paradigm.

Neural gas (NG) as proposed in (Martinetz et al. 1993) is a particularly robust variation of vector quantization with the introduction of neighborhood relations. The NG system takes into account the relative distances between all prototypes and a given pattern and *ranks* the prototypes accordingly. The rank-based adaptation steps are hence affected directly by the data topology. This is contrast to the self-organizing map (Kohonen 1997) which utilizes a predefined lattice. While the NG procedure is potentially more expensive than SOM, it also reduces topology mismatches.

In practice, given proper choices of the learning parameters, NG algorithms yield better solutions than WTA; however, the effect of this strategy on convergence speed or asymptotic behavior has hardly been rigorously investigated so far.

Methods from statistical physics and the theory of on-line learning (Engel and van den Broeck 2001) allow for an exact mathematical description of learning systems for high dimensional data. In the limit of infinite dimensionality, such systems can be fully described in terms of a few characteristic quantities, the so-called *order parameters*. The evolution of these order parameters along the training procedure is characterized by a set of coupled ordinary differential equations (ODE). By integrating these ODEs, it is possible to analyse the performance of VQ algorithms in terms of stability, sensitivity to initial conditions, and achievable quantization error. This successful approach has also been reviewed in (Engel and van den Broeck 2001, Watkin et al. 1993), among others.

The extension of the theoretical analysis of simple (WTA-based) vector quantization with two prototypes and two clusters introduced in an earlier work (Biehl et al. 2006) is not straightforward. Additional prototypes and clusters introduce more complex interactions in the system that can result in radically different behaviors. Also, the mathematical treatment becomes more involved and requires, for instance, several numerical integrations. In this work we introduce an additional prototype and a mixture of clusters. We investigate not only WTA but also the popular Neural Gas approach (Martinetz et al. 1993) for vector quantization. This is an important step towards the investigation of general VQ approaches based on neighborhood interaction such as self-organizing maps.

3.2 Winner-Takes-All and Neural Gas

Assume a set of P input data denoted by $\{\xi^\mu \in I\!R^N\}_{\mu=1}^P$, generated according to a given probability density function $P(\xi)$. In our work, we choose the input density to be a mixture of M spherical Gaussian clusters, as written in Eq. (2.5):

$$P(\xi)=\sum_{\sigma=1}^{M} p_\sigma P(\xi|\sigma) \text{ with } P(\xi|\sigma) = \frac{1}{(2\pi\upsilon_\sigma)^{N/2}} \exp\left(-\frac{1}{2\upsilon_\sigma}(\xi - \ell_\sigma \mathbf{B}_\sigma)^2\right) \quad (3.1)$$

where p_σ are the prior probabilities of each cluster. The components of vector ξ^μ are random numbers according to a Gaussian distribution with mean vectors $\ell_\sigma \mathbf{B}_\sigma$ and variance υ_σ. The mean vectors are orthogonal, i.e. $\mathbf{B}_i \cdot \mathbf{B}_j = \delta_{ij}$ where δ is the Kronecker delta. The parameters ℓ_σ describe the separation between the clusters. We refer to the details of this input density in Chapter 2.2.

Vector Quantization represents the input data in the same N-dim. space by a set of prototypes $\mathbf{W} = \left\{\mathbf{w}_S \in I\!R^N\right\}_{S=1}^K$. The primary goal of VQ is to find a faithful representation by minimizing the so-called quantization or distortion error

$$E(\mathbf{W}) \quad = \quad \frac{1}{2}\sum_{\mu=1}^{P}\sum_{S=1}^{K} d(\xi^\mu, \mathbf{w}_S) \prod_{T\neq S} \Theta_{ST} - \frac{1}{2}\sum_{\mu=1}^{P}(\xi^\mu)^2 \quad (3.2)$$

where $\Theta_{TS} \equiv \Theta\big(d(\xi^\mu, \mathbf{w}_S) - d(\xi^\mu, \mathbf{w}_T)\big)$. For each input vector ξ^μ the closest prototype $\mathbf{w}_S$ is singled out by the product of Heaviside functions, $\Theta(x) = 0$ if $x < 0$; 1 else. Here we restrict ourselves to the quadratic Euclidean distance measure $d(\xi^\mu, \mathbf{w}_S) = (\xi^\mu - \mathbf{w}_S)^2$. The constant $\frac{1}{2}\sum_{\mu=1}^{P}(\xi^\mu)^2$ term is independent of prototype positions and is subtracted for convenience, this will be shown in Eq. (3.17).

The input data is presented sequentially during training and one or more prototypes are updated on-line. Algorithms studied here can be interpreted as stochastic gradient descent procedures with respect to a cost function $H(\mathbf{W})$ related to $E(\mathbf{W})$. The generalized form reads

$$H(\mathbf{W}) \quad = \quad \frac{1}{2}\sum_{\mu=1}^{P}\sum_{S=1}^{K} d(\xi^\mu, \mathbf{w}_S) f(r_S) - \frac{1}{2}\sum_{\mu=1}^{P}(\xi^\mu)^2 \quad (3.3)$$

where r_S is the rank of prototype $\mathbf{w}_S$ with respect to the distance $d(\xi^\mu, \mathbf{w}_S)$, i.e. $r_S = S - \sum_{T\neq S}\Theta_{ST}$. Rank $r_J = 1$ corresponds to the so-called *winner*, i.e. the prototype $\mathbf{w}_J$ closest to the example ξ^μ. The rank function $f(r_S)$ determines the update strength for the set of prototypes and satisfies the normalization $\sum_{S=1}^{K} f(r_S) = 1$; note that it does not depend explicitly on distances but only on the ordering of the prototypes with respect to the current example.

Stochastic gradient descent of $H(\mathbf{W})$ yields the online update rule

$$\mathbf{w}_S^\mu = \mathbf{w}_S^{\mu-1} + \frac{\eta}{N} f(r_S)(\xi^\mu - \mathbf{w}_S^{\mu-1}), \tag{3.4}$$

where η is the learning rate and ξ^μ is a single example drawn independently at time step μ of the sequential training process. We compare two different algorithms:

1. **Winner Takes All**:
   In this learning scheme only one prototype, the winner, is updated for each input. The cost function directly minimizes the quantization error with $H(\mathbf{W}) = E(\mathbf{W})$. The corresponding rank function is

$$f_{\mathrm{WTA}}(r_S) = \prod_{T \neq S}^{K} \Theta_{ST}. \tag{3.5}$$

2. **Neural Gas**:
   The update strength decays exponentially with the rank controlled by a parameter λ. The rank function is $f(r_S) = \frac{1}{C(\lambda)} h_\lambda(r_S)$ where $h_\lambda(.) = \exp(-r_S/\lambda)$ and $C(\lambda) = \sum_{r_S=1}^{K} \exp(-r_S/\lambda)$ is a normalization constant. The parameter λ is adjusted during training; it is frequently set large initially and decreased in the course of training. Note that for $\lambda \to 0$ the NG algorithm becomes identical with WTA. We decompose $f(r_S)$ according to its ranks as

$$f_{\mathrm{NG}}(r_S) \quad = \quad \frac{1}{C(\lambda)} \sum_{k=1}^{K} h_\lambda(k) g_S(k) \tag{3.6}$$

   where $g_S(k) = 1$ if $r_S = k$; 0 else and $\sum_k g_S(k) = 1$. In a system of three prototypes, this can be written in terms of Heaviside functions, defined below Eq. (3.2):

$$\begin{aligned} g_S(1) &= \prod_{T \neq S}^{K} \Theta_{ST} \\ g_S(2) &= \sum_{U \neq S}^{K} \prod_{T \neq U,S} \Theta_{ST}(1 - \Theta_{SU}) \\ g_S(3) &= \prod_{T \neq S}^{K} \big(1 - \Theta_{ST}\big). \end{aligned} \tag{3.7}$$

3.3 Analysis of learning dynamics

In this section we give a description of the theoretical framework for analysis of on-line LVQ training. Following the lines of the theory of on-line learning, e.g. (Biehl and Caticha 2003, Biehl et al. 2004, Biehl et al. 2007, Engel and van den Broeck 2001, Saad 1999), in the thermodynamic limit $N \to \infty$ the system can be fully described in terms of a few characteristic quantities, or so-called order parameters. A suitable set of characteristic quantities for the considered learning model is:

$$R_{S\sigma}^{\mu} = \mathbf{w}_S^{\mu} \cdot \mathbf{B}_{\sigma} \qquad Q_{ST}^{\mu} = \mathbf{w}_S^{\mu} \cdot \mathbf{w}_T^{\mu}. \tag{3.8}$$

Note that $R_{S\sigma}$ measure the projections of prototype vectors $\mathbf{w}_S^{\mu}$ on the center vectors $\mathbf{B}_{\sigma}$ and Q_{ST}^{μ} correspond to the self- and cross- overlaps of the prototype vectors.

From the generic update rule defined above, Eq. (3.4), we can derive the following recursions in terms of the order parameters:

$$\begin{aligned} \frac{R_{S\sigma}^{\mu} - R_{S\sigma}^{\mu-1}}{1/N} &= \eta f(r_S)\big(b_{\sigma}^{\mu} - R_{S\sigma}^{\mu-1}\big) \\ \frac{Q_{ST}^{\mu} - Q_{ST}^{\mu-1}}{1/N} &= \eta\big[f(r_T)\big(h_S^{\mu} - Q_{ST}^{\mu-1}\big) + f(r_S)\big(h_T^{\mu} - Q_{ST}^{\mu-1}\big)\big] \\ &\quad + \eta^2 \frac{f(r_S)f(r_T)(\xi^{\mu})^2}{N} + \mathcal{O}\left(\frac{1}{N}\right) \end{aligned} \tag{3.9}$$

where the input data vectors ξ^{μ} enter the system only as their projections h_S^{μ} and b_{σ}^{μ}, defined as

$$h_S^{\mu} = \mathbf{w}_S^{\mu-1} \cdot \xi^{\mu} \qquad b_{\sigma}^{\mu} = \mathbf{B}_{\sigma} \cdot \xi^{\mu}. \tag{3.10}$$

Note that the last two terms in Eq. (3.9) come from

$$\begin{aligned} &\left(\frac{\eta}{N}\right)^2 f(r_S)f(r_T)(\xi^{\mu} - \mathbf{w}_S^{\mu-1})(\xi^{\mu} - \mathbf{w}_T^{\mu-1}) \\ &\qquad = \frac{\eta^2}{N}\frac{f(r_S)f(r_T)\left[(\xi^{\mu})^2 - h_S^{\mu} - h_T^{\mu} + Q_{ST}^{\mu-1}\right]}{N} \end{aligned} \tag{3.11}$$

where $(\xi^{\mu})^2$ is the only term that scales with N.

In the limit $N \to \infty$, the $\mathcal{O}(1/N)$ term can be neglected and the order parameters become *self -averaging* with respect to the random sequence of examples. This means that fluctuations of the order parameters vanish and the system dynamics can be described exactly in terms of their mean values. A mathematically rigorous analysis of this property is provided in (Reents and Urbanczik 1998), describing the necessary bounds of the magnitude of the learning steps.

We denote the average over the density $P(\xi)$ as $\langle\cdots\rangle = \sum_\sigma p_\sigma \langle\cdots\rangle_\sigma$, where $\langle\cdots\rangle_\sigma$ is the conditional average over $P(\xi|\sigma)$, and exploit the thermodynamic limit to find the following relation

$$\begin{aligned}\lim_{N\to\infty}\frac{\langle f(r_S)f(r_T)\xi^2\rangle}{N} &= \lim_{N\to\infty}\frac{1}{N}\sum_\sigma p_\sigma \left\langle f(r_S)f(r_T)\left\langle\xi^2\right\rangle_\sigma\right\rangle_\sigma \\ &= \sum_\sigma p_\sigma v_\sigma \left\langle f(r_S)f(r_T)\right\rangle_\sigma . \end{aligned} \tag{3.12}$$

Here we used the relation $\xi^2 = \left\langle\xi^2\right\rangle_\sigma + \Gamma$, where Γ is a stochastic quantity with $\mathcal{O}(\Gamma) \ll N$ which can be neglected. Also, we calculated the conditional average of the input vector length $\left\langle\xi^2\right\rangle_\sigma /N = (v_\sigma N + \ell_\sigma^2)/N$, see the derivations in Eq. (2.6).

Furthermore, for $N \to \infty$, we can conceive learning time as a continuous variable by using the rescaled quantity

$$\alpha \equiv \mu/N. \tag{3.13}$$

Here it is implied that successful training requires a number of examples which grows linearly with the number of dimensions. Accordingly, by combining Eqs. (3.9), (3.12) and rescaling with (3.13), the dynamics can be described by a set of coupled ODE after performing an average over the sequence of input data, see also (Biehl et al. 2004, Ghosh et al. 2006) :

$$\begin{aligned}\frac{dR_{S\sigma}}{d\alpha} &= \eta\big(\langle b_\sigma f(r_S)\rangle - \langle f(r_S)\rangle R_{S\sigma}\big) \\ \frac{dQ_{ST}}{d\alpha} &= \eta\big(\langle h_S f(r_T)\rangle - \langle f(r_T)\rangle Q_{ST} + \langle h_T f(r_S)\rangle - \langle f(r_S)\rangle Q_{ST}\big) \\ &\quad +\eta^2\sum_\sigma p_\sigma v_\sigma \left\langle f(r_S)f(r_T)\right\rangle_\sigma . \end{aligned} \tag{3.14}$$

In various sections in this thesis, we investigate learning behaviors using small learning rates $\eta \to 0$ and neglect the η^2 terms in Eq. (3.14). Non trivial behavior is only expected by rescaling the learning time while taking the simultaneous limits

$$\eta \to 0,\ \alpha \to \infty,\ \widetilde{\alpha} = \eta\alpha.$$

In the limit of small learning rates, Eq. (3.14) is rescaled as

$$\begin{aligned}\frac{dR_{S\sigma}}{d\widetilde{\alpha}} &= \langle b_\sigma f(r_S)\rangle - \langle f(r_S)\rangle R_{S\sigma} \\ \frac{dQ_{ST}}{d\widetilde{\alpha}} &= \langle h_S f(r_T)\rangle - \langle f(r_T)\rangle Q_{ST} + \langle h_T f(r_S)\rangle - \langle f(r_S)\rangle Q_{ST}. \end{aligned} \tag{3.15}$$

Exploiting the limit $N \to \infty$ once more, the quantities h_S^μ, b_σ^μ in Eq. (3.14) or Eq. (3.15) become correlated Gaussian quantities by means of the Central Limit Theorem. Therefore, they are fully specified by first and second moments, detailed in Appendix A:

$$\begin{aligned} &\langle h_S^\mu \rangle_\sigma = \ell_\sigma R_{S\sigma}^{\mu-1}, \quad \langle b_\tau^\mu \rangle_\sigma = \ell_\sigma \delta_{\tau\sigma}, \quad \langle h_S^\mu h_T^\mu \rangle_\sigma - \langle h_S^\mu \rangle_\sigma \langle h_T^\mu \rangle_\sigma = \upsilon_\sigma Q_{ST}^{\mu-1} \\ &\langle b_\tau^\mu b_\rho^\mu \rangle_\sigma - \langle b_\tau^\mu \rangle_\sigma \langle b_\rho^\mu \rangle_\sigma = \upsilon_\sigma \mathbf{T}_{\tau\rho}, \quad \langle h_i^\mu b_\tau^\mu \rangle_\sigma - \langle h_i^\mu \rangle_\sigma \langle b_\tau^\mu \rangle_\sigma = \upsilon_\sigma R_{i\tau}^{\mu-1} \,. \end{aligned} \tag{3.16}$$

where S, T are prototype indices, τ, ρ, σ are cluster indices, δ is the Kronecker delta and $\mathbf{T}_{\tau\rho} \equiv \mathbf{B}_\tau \cdot \mathbf{B}_\rho$ is an overlap measure between clusters. For orthonormal $\mathbf{B}_\sigma$ vectors, we can use $\mathbf{T}_{\tau\rho} = 1$, if $\tau = \rho$; 0 else.

Given the averages for a specific rank function $f(r_S)$, cf. Eqs. (B.38) and (B.46) we obtain a closed form expression of ODE. Using the initial conditions $R_{S\sigma}(0)$, $Q_{ST}(0)$, we integrate this system for a given algorithm and get the evolution of order parameters in the course of training, $R_{S\sigma}(\alpha), Q_{ST}(\alpha)$. The behavior of the system depends on the characteristic of the data and the parameters of the learning scheme, i.e. offset of the clusters ℓ_σ, variance within the clusters υ_σ, learning rate η, and for NG, the rank function parameter λ. This method of analysis is in good agreement with large scale Monte Carlo simulations of the same learning systems for dimensionality as low as $N = 200$, see e.g. in (Biehl et al. 2007).

Analogously, the average quantization error, Eq. (3.2), over the probability density expressed in terms of order parameters

$$E(\mathbf{W}) = \frac{1}{2} \sum_{S=1}^{K} \left\langle \prod_{T \neq S}^{K} \Theta_{ST} \right\rangle Q_{SS} - \sum_{S=1}^{K} \left\langle h_S \prod_{T \neq S}^{K} \Theta_{ST} \right\rangle \tag{3.17}$$

Note that $E(\mathbf{W})$ does not depend explicitly on ξ; here it is shown how the subtracted constant term described in Eq. (3.2) and Eq. (3.3) becomes useful. $E(\mathbf{W})$ is fully expressed in terms of order parameters. For instance, in two prototype systems $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2\}$ we can calculate the above quantity $E(\mathbf{W}) = \sum_\sigma p_\sigma E_\sigma(\mathbf{W})$ as follows:

$$E_\sigma(\mathbf{W}) \;=\; -\frac{\sqrt{\upsilon_\sigma}\Delta Q}{\sqrt{2\pi}} \exp\left(-\frac{Z^2}{2}\right) + \left(\frac{Q_{11}}{2} - \ell_\sigma R_{1\sigma}\right)\Phi(Z) + \left(\frac{Q_{22}}{2} - \ell_\sigma R_{2\sigma}\right)\Phi(-Z)\,,$$

$$\begin{aligned} \text{with} \qquad & \Delta Q = \sqrt{Q_{11} - 2Q_{12} + Q_{22}}\,, \\ & Z = \left(2\ell_\sigma(R_{1\sigma} - R_{2\sigma}) - Q_{11} + Q_{22}\right) / \left(2\sqrt{\upsilon_\sigma}\Delta Q\right). \end{aligned} \tag{3.18}$$

We refer to the details of the calculations in Appendix B.4. The form of $E_\sigma(\mathbf{W})$ for systems with more prototypes is more involved, and requires numerical integrations over a $(K-2)$ subspace. Plugging in the values of the order parameters $\{R_{S\sigma}(\alpha), Q_{ST}(\alpha)\}$, we can study the so-called learning curve $E(\mathbf{W})$ in dependence of the training time α for a given VQ algorithm.

3.4 Results

3.4.1 Learning Dynamics

We study the performance of both WTA and NG in several cases using three prototypes and up to three clusters. Stochastic gradient descent procedures approach a (local) minimum of the objective function in the limit $\eta \to 0$. We can consider this limit exactly by rescaling the learning time as $\widetilde{\alpha} = \eta\alpha$. Then, the $\mathcal{O}(\eta^2)$ terms in Eq. (3.14) can be neglected and the set of ODEs is simplified. For all demonstrations, the NG algorithm is studied for decreasing λ with

$$\lambda(\widetilde{\alpha}) = \lambda_i(\lambda_f/\lambda_i)^{\widetilde{\alpha}/\widetilde{\alpha}_f} \tag{3.19}$$

where λ_i and λ_f are respectively the initial and final settings of the rank parameter and $\widetilde{\alpha}_f$ is a learning time parameter. The influence of the initial set of prototypes on the learning curves is investigated by choosing different values of $\{R_{S\sigma}(0), Q_{ST}(0)\}$.

Figure 3.1 presents the prototype dynamics in a system with three prototypes

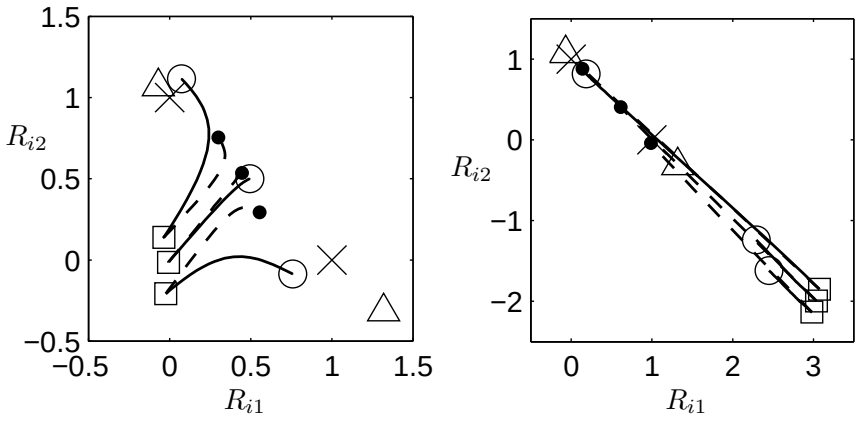

Figure 3.1: *Trajectories of prototypes* $\{\mathbf{w}_i\}_{i=1}^3$ *on the plane spanned by* $\mathbf{B}_1$ *and* $\mathbf{B}_2$*, displayed with the order parameters* $R_{i\sigma} = \mathbf{w}_i \cdot \mathbf{B}_\sigma$*. The cluster centers* $\ell_\sigma \mathbf{B}_\sigma$ *are marked by crosses. The trajectories are marked by solid lines (WTA) and dashed lines (NG). The prototypes at initialization are marked with squares and at* $\widetilde{\alpha} = 10$ *with circles (WTA) and dots (NG). Both algorithms converge at the triangles, where two prototypes coincide at* $\{-0.07, 1.07\}$*. The set of prototypes is initially set (a) near the cluster centers, and (b) far away from the cluster centers. In both figures the parameters are* $p_1 = 0.45$*,* $\ell_1 = \ell_2 = 1$*,* $\upsilon_1 = \upsilon_2 = 1$*,* $\eta \to 0$*,* $\lambda_i = 2$*,* $\lambda_f = 0.01$ *and* $t_f = 50$*.*

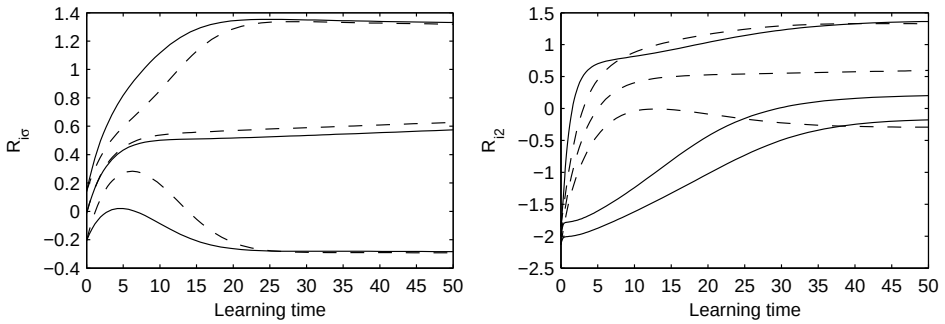

Figure 3.2: *The corresponding order parameters R_{i2} at learning time $\widetilde{\alpha} = \eta\alpha$ for WTA (solid lines) and NG (dashed lines) algorithms in the system described in Fig.3.1. The initial sets of prototypes are defined in Figs.3.1(a) and (b), respectively.*

and two clusters. We examine two different initial sets of prototypes: close to the origin at $\{R_{S1}(0), R_{S2}(0)\} \approx \{0, 0\}, Q_{ST}(0) \approx 0, \forall\{S, T\}$ in Fig. 3.1; and far away from the origin on the side of the weaker cluster, viz. p_1, at $\{R_{S1}(0), R_{S2}(0)\} \approx \{3, -2\}$, $Q_{ST}(0) = R_{S\sigma}(0) \cdot R_{T\sigma}(0), \forall\{S, T\}$ in Fig. 3.1. While the prototypes have different trajectories in WTA and NG algorithms, they converge at the identical configuration at large α and $\lambda \to 0$. Here, the projections of two prototypes converge near the center of the stronger cluster. The advantage of NG is apparent in Fig. 3.1 where all prototypes already reach the area near the cluster centers at an intermediate learning stage $\widetilde{\alpha} = 10$.

This can be illustrated with the evolution of the order parameters $R_{S2}(\alpha)$ in Fig. 3.2. In Fig. 3.2, the order parameters of both algorithms converge relatively fast. In Fig. 3.2, the order parameters of one prototype change rapidly compared to that of other prototypes in WTA algorithm. One prototype dominates as the winner and gets frequent updates towards the cluster centers, while the other prototypes are rarely updated. The NG algorithm partially solves this problem by updating all prototypes at the initial stages of learning.

The quantization error obtained from the order parameters $\{R_{S\sigma}(\widetilde{\alpha}), Q_{ST}(\widetilde{\alpha})\}$ is displayed in Fig. 3.3. We observe that the quantization error decreases faster in the WTA algorithm compared to NG methods at the initial stages of the learning. This behavior can be explained by the fact that the H_{NG} differs from $E(\mathbf{W})$ by smoothing terms in particular in early stages of training. We observe that WTA yields the best overall quantization error in the first set of initial values in Fig. 3.3. This is mirrored by the fact that, for large $\widetilde{\alpha}$ and $\lambda_f \to 0$, both algorithms yield the same quantization error.

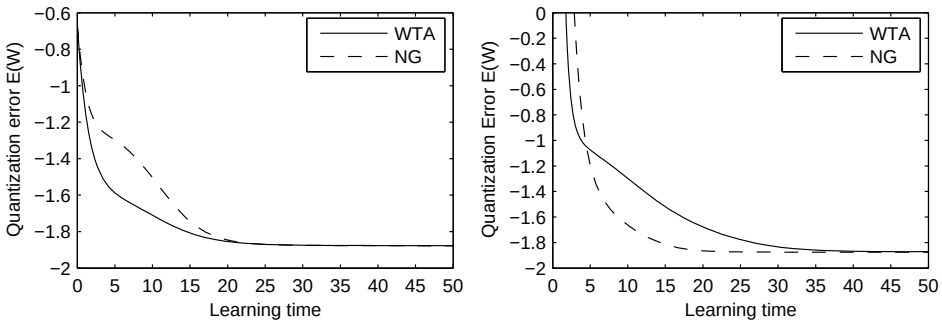

Figure 3.3: *Evolution of the quantization error $E(\mathbf{W})$ in Fig. 3.1 at learning time $\widetilde{\alpha} = \eta\alpha$ for WTA (solid line) and NG (dashed line) algorithms. The prototypes are initialized (a) near the cluster centers and (b) far away from the cluster centers.*

For WTA training, the prototypes reach $\widetilde{\alpha} \to \infty$ asymptotic positions corresponding to the global minimum of $E(\mathbf{W})$ for small learning rates $\eta \to 0$. However, learning can slow down significantly at intermediate stages of the training process. Transient configurations may persist in the vicinity of local minima and can indeed dominate the training process. The NG is more robust w.r.t. the initial position of prototypes than WTA while achieving the best quantization error asymptotically.

3.4.2 Asymptotic configuration

The dynamics of the prototypes while learning on a model data with a larger separation between the clusters are presented in Fig. 3.4. The initial configurations correspond to the following values of $\{R_{S1}(0), R_{S2}(0)\}$: (a) $\{-1, 2\}$, (b) $\{-0.5, 2\}$, (c) $\{-1, 1.5\}$, (d) $\{-0.5, 1.5\}$, (e) $\{-1, 1\}$ and (f) $\{-0.5, 1\}$. In all panels, $Q_{ST}(0) = R_{S\sigma}(0) \cdot R_{T\sigma}(0)$.

In this case, the optimal configuration of prototypes is with two prototypes representing the stronger cluster as in Figs. 3.4(a to c). However, the asymptotic configuration of the prototypes in the WTA algorithm are sensitive to the initial conditions. In some cases, viz. Figs. 3.4(d to f), this configuration is not the optimal set of prototypes. Therefore, even in this comparably simple model, prototypes in WTA can be confined in suboptimal local minima of the cost function $E(\mathbf{W})$. The issue of different regions of initialization which lead to different asymptotic configurations are to be discussed in forthcoming projects.

The asymptotic configurations for the NG algorithm are independent of initial conditions as shown in Figs. 3.4(a to f). During the learning process with $\lambda > 0$ the

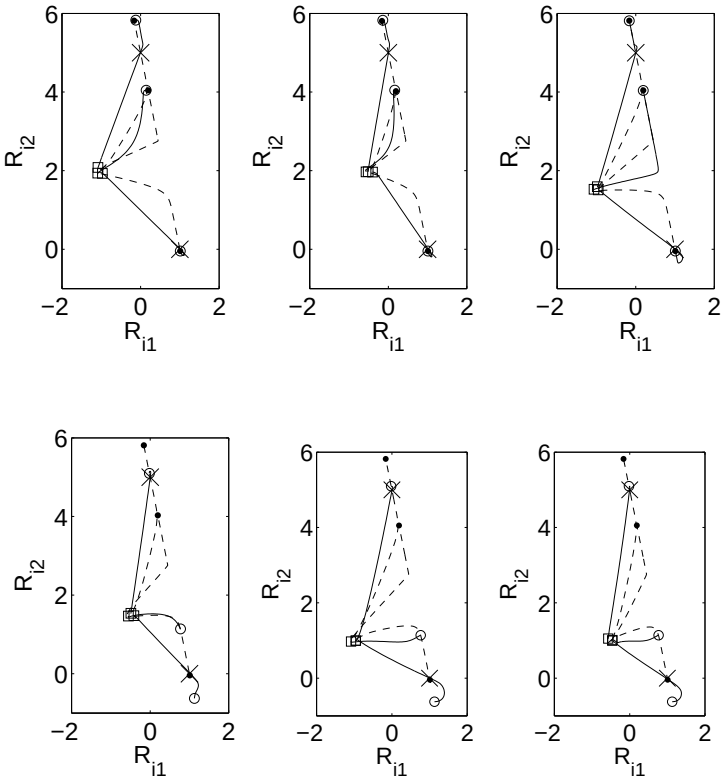

Figure 3.4: *Trajectories of the prototypes on the plane spanned by* $\mathbf{B}_1$ *and* $\mathbf{B}_2$*, corresponding to the WTA (solid lines) and the NG (dashed lines) algorithms. Here,* $p_1 = 0.45$*,* $p_2 = 0.55$*,* $\upsilon_1 = 1$*,* $\upsilon_2 = 1.21$*,* $\ell_1 = 1$ *and* $\ell_2 = 5$*. The cluster centers* $\ell_\sigma \mathbf{B}_\sigma$ *are marked by* $\times$*. The initial prototype configurations for both algorithms are marked with* $\square$*. While the asymptotic configurations of WTA (circles) algorithm depends on initialization, the NG (dots) always produces identical asymptotic configurations. In these cases, the NG algorithm always finds the optimal quantization error.*

system moves towards intermediate configurations with minimum $H_{\mathrm{NG}}(\mathbf{W})$. Given sufficiently large λ and $\widetilde{\alpha}$, these configurations are identical and therefore the NG algorithm is robust with respect to initial conditions. In these cases, the asymptotic configuration is the optimal configuration and thus the NG algorithm achieves optimal performance.

We demonstrate a model where the NG algorithm does not yield optimal per-

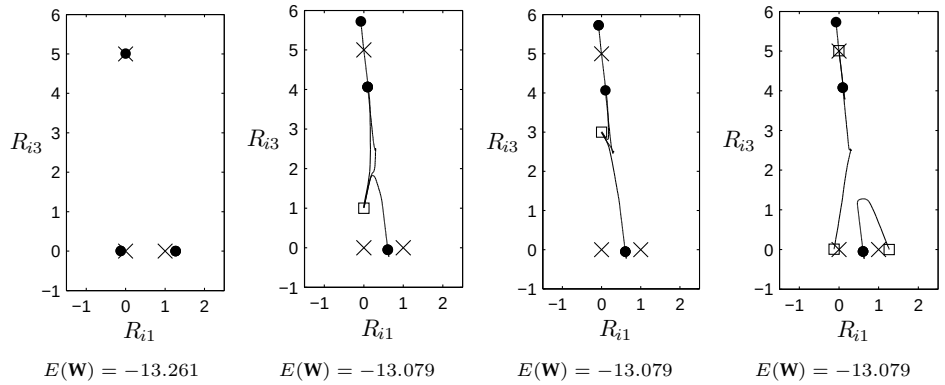

Figure 3.5: *(a) The optimal set of prototypes (solid dots) in a system with three clusters projected on the plane space spanned by* $\{\mathbf{B}_1, \mathbf{B}_3\}$. *The values of* R_{i2} *are not shown here. The cluster centers* $\ell_\sigma \mathbf{B}_\sigma$ *are marked by* $\times$. *(b,c,d) Trajectories of the prototypes using the NG algorithm with different initial conditions. Their initial (squares) and asymptotic (solid dots) configuration of the prototypes are indicated. The parameters are* $p_1 = 0.25, p_2 = 0.20, p_3 = 0.55$, $\upsilon_1 = \upsilon_2 = \upsilon_3 = 1$, $\ell_1 = \ell_2 = 1$ *and* $\ell_3 = 5$.

formance in Fig. 3.5. In this more complex situation, the weaker cluster ($p_\sigma = 0.45$) is divided into two Gaussian clusters with $p_{1,2} = \{0.25, 0.20\}$. This corresponds to a system of three clusters, with $\ell_\sigma = \{1, 1, 5\}$ and $p_\sigma = \{0.25, 0.20, 0.55\}$. The distance between the first two clusters is small compared to their distance to the third cluster. In comparison to the previous case, where the weaker cluster spreads out evenly in all directions, here it has a particular orientation along the vector $(\mathbf{B}_1 - \mathbf{B}_2)$. Because of this structure, the best quantization error is obtained when one prototype is placed near each cluster center, as in Fig. 3.5, even though one cluster has a very large prior ($p_3 > p_1 + p_2$).

Similar to the previous case, the asymptotic configuration for the NG algorithm is independent of initial conditions. However, this configuration with two prototypes near the center of the stronger cluster in Figs. 3.5(b to d), is not the optimal configuration. Even with prototypes initialized at the optimal set as in Fig. 3.5, the NG algorithm may still lead to suboptimal configurations.

The characteristics of the cost function $H(\mathbf{W})$ of NG, ie. its minima, can be radically apart with different values of λ. While the NG may find the configuration of the global minima of $H(\mathbf{W})$ for large λ, these configurations do not always lead to the global minima for smaller λ. Consequently, the asymptotic configuration may correspond to a local minimum of $E(\mathbf{W})$ and the NG algorithm does not always

yield the optimal quantization error.

3.5 Conclusion

We have presented an exact mathematical analysis of the dynamics of vector quantization for high dimensional data. Performance is measured by the evolution of the quantization error. In a learning scenario with no sub-optimal local minima of the quantization error, the WTA always converges to the best quantization error. However, learning can slow down significantly if the prototypes are initialized far from the region of high data density. The NG is less sensitive to the initial conditions and achieves both robustness and optimal asymptotic quantization error. Thereby, the convergence speed of NG algorithms is comparable or (for initialization outside the clusters) better than the convergence speed of simple WTA mechanisms, while achieving the same final quantization error.

In the presence of local minima, the WTA algorithm may converge into different asymptotic configurations depending on its initial conditions. The NG algorithm is very robust, i.e. relatively insensitive to initial conditions. However, we demonstrate a test case where it does not find the best asymptotic quantization error. The above discussed sub-optimal outcome of NG training might result from the specific schedule at which λ is decreased in the course of training. The influence of both schedules for η and λ will be studied in greater detail in forthcoming projects.

The formalism allows for the design of optimal schemes in the framework of the model situation. While this model clearly does not describe the complexity of real world problems, it is useful to demonstrate certain characteristics of both algorithms. Immediate extensions of the model towards realistic data structures could include additional or non-spherical clusters. An important investigation for rank-based learning schemes could include the analysis of the popular Self Organising Maps (SOM) schemes, which apply a predefined lattice of prototype. Comparisons of SOM to Neural Gas systems would provide insight on the importance of preserving the topology of prototypes.